



Your Text Here

# Efficient Organization For Production Processing

Anna Mazzacane on behalf of OPOS Team

FIFE Workshop

20<sup>th</sup>-21<sup>th</sup> June 2016

# Introduction

- Is an efficient production processing possible?
- In this presentation I will give a list of **known** practices and procedure to use resources wisely and to avoid this:



- I will also present how to setup the keepup jobs for daily processing of data.
- I assume you are familiar with the production tools. If not, this workshop is a great opportunity.

# Checklist for an efficient production processing

- The first important step is to design an efficient workflow for production processing.
  - ✓ This task needs a good knowledge of the infrastructure and tools available.
  - ✓ Take the advantage to contact CD experts. They can provide good practice tips and advice the most effective configuration for the infrastructure according to experiment needs.
  - ✓ See all “Best Practices” talks at this workshop.
  
- Design the workflow in order to be able to run on the OSG and extend grid resources.
  - ✓ There are many resource available and free to use on the OSG.  
The SCD is working to make their use easy and transparent to use.
  - ✓ Experiments needs to extend their configuration to be able to use OSG resources.
  - ✓ See Tanya’s talk: Access to Off-Site Resources.
  - ✓ See Ken’s talk: OSG Submission Tutorial.
  
- Use common tools like SAM for data management, jobsub for the job submission, ifdch to handle data transfer and expand manpower capability while taking advantage from others experience.
  - ✓ Jobsub is an integrated submission system to easily submit batch jobs to the GRID.
  - ✓ SAM is a powerful tool to handle data files.  
Create dataset, monitor the status of the files, in general automatize part of the production workflow and make the ”productioners” life easy.
  - ✓ See Robert’s talk: Best Practices: SAM.
  - ✓ See Dennis’ talk: Best Practices: Jobsub.

# Checklist for an efficient production processing

- Keep production tools updated to the latest version to avoid painful and difficult late alignment.
  - ✓ The use of the latest version of production tools guarantee the access to new features and improvements.
  - ✓ Testing the pre-release, as asked through the the computing liasons of your experiment will guarantee that the tools is properly working for you and it is what you need.
- Avoid hard coded settings, use configuration files instead with the list of options in order to avoid to make “many changes in many places”.
  - ✓ Hard coded settings make needed changes difficult to apply and can cause issues issue hard to track back.
- Production code, scripts and configuration files should be tagged with a supported version control system.
  - ✓ Assign a release tag or global version (as appropriate for the version control system being used) captures the complete state of the system used for a submission.
  - ✓ Config files/scripts for production should be contained within a package or set of packages/modules which names can distinguish them as production level content, like “ProductionScripts”

# Checklist for an efficient production processing

- Check the right permission to run jobs on the grid.
  - ✓ Add some check in the submission script to be sure having valid credentials to get the jobs submitted, to be able to access the input files, to have write permission for the output directory
  - ✓ Submit few test jobs before the submission of hundreds of jobs on the grid.
  - ✓ Or better:
  
- Use the Continuous Integration (CI) system to check the production workflow is healthy and working.
  - ✓ The CI practice provides rapid feedback to help identify “bugs” introduced by code changes as soon as possible, *hopefully* before the code goes in production. Issues detected early on in development are typically smaller, less complex and easier to resolve.
  - ✓ See Vito’s talk: [Continuous Integration Project](#).
  
- Monitor production jobs using existing tools, like fifemon, fts monitoring, POMS
  - ✓ SCD/FIFE is providing plenty of tools to monitor each stage of the production.
  - ✓ FIFEmon provides dashboards to monitor job status, job efficiency onsite and offsite; grid status and much much more.
  - ✓ FTS monitoring provides information about the files.
  - ✓ SAM monitoring provides information about the data being processed.
  - ✓ POMS (in development phase) allows to schedule jobs submission and the recovery of failed jobs; monitors the status of the jobs and files processed by each job and helps debugging errors through the parsing of jobs log.
  - ✓ See Marc’s talk: [POMS](#).
  - ✓ See Kevins’ tutorial: [FIFEmon](#).

# Keepup jobs and user credentials

- Keepup jobs are intended to process data as they are produced.
- You can schedule those jobs using a **crontab**.
- The crontab execute a **cronjob** that can run a script to select the data and to submit the jobs.
- Example of cronjob:

```
30 01 * * * /script/to/execute.sh
```

(the first two fields are the time and the latest three are the date)

The time and date fields are:

| field | allowed values |
|-------|----------------|
|-------|----------------|

|        |      |
|--------|------|
| minute | 0-59 |
|--------|------|

|      |      |
|------|------|
| hour | 0-23 |
|------|------|

|              |      |
|--------------|------|
| day of month | 1-31 |
|--------------|------|

|       |                 |
|-------|-----------------|
| month | 1-12 (or names) |
|-------|-----------------|

|             |                                   |
|-------------|-----------------------------------|
| day of week | 0-7 (0 or 7 is Sun, or use names) |
|-------------|-----------------------------------|

- To process those jobs you need a valid proxy with role Production.
- The proxy has a limited life time and must be periodically renewed.
- To automatically renew the proxy you need another cronjob.

## To summarize:

- To setup keepup jobs you need:
  - ✓ A cronjob to automatically submit the jobs
  - ✓ A cronjob to automatically renew the proxy
- In the cronjob you schedule the time and the date to execute a task.
- The task can be command lines or a script.

# Keepup jobs and user credentials (cont'd)

- So far so good, BUT ...
- To generate a proxy you need a certificate.
- To have a certificate you need a kerberos ticket.
- To have a kerberos ticket you need to type your passw.

**This is not what you want in an automatic procedure!**

- You can use the following procedure:
  - ✓ Create a **keytab** and a **cron principal**.
  - ✓ Create a **production proxy** using the cron principal.
  - ✓ Open a SNOW request to add the subject of this proxy to your voms user credentials.
  - ✓ Add your cron principal in your .k5login
- You can now setup the cronjob that:
  - ✓ Uses your keytab to get a kerberos ticket
  - ✓ Get a certificate
  - ✓ Renew the production proxy
  - ✓ Copy the production proxy in a local dir where you will submit the jobs.

You can find the procedure in the backup slides or at the following redmine page:

[https://cdcvns.fnal.gov/redmine/projects/nova-production/wiki/Keepup\\_Jobs](https://cdcvns.fnal.gov/redmine/projects/nova-production/wiki/Keepup_Jobs)

- Use this production proxy to submit production jobs using your user account.
- Contact OPOS team to setup the procedure to get the environment configured for the productioner shifters.

# Summary

- IF experiments are very big data producers.
- The production processing needs to be very efficient. Resources are not unlimited.
- Performance, stability and usability are the primary factors:
  - ✓ The infrastructure must be able to guarantee access and functionalities. (Not covered in this talk).
  - ✓ The production workflow must use the infrastructures well.
- Many tools have been developed and tested by other experiments: take advantage of others experience.
- Many variables are involved in this process, not easy.
- Presented what to check for efficient production processing and the procedure to run keepup jobs using user credential.
- To achieve the goal, collaboration, cooperation and communication are the key.



- **To create the keytab, the cron principal and the production proxy using the cron principle:**
  - ✓ Login as yourself in gpvm01.fnal.gov
  - ✓ Type kcroninit
    - Return
    - Return
    - Input Kerberos principal
    - Input kerberos password
  - ✓ Type kcron
  - ✓ Type kx509
  - ✓ Type voms-proxy-info -all
    - Get the subject of that output, it should look like:  
subject= /DC=gov/DC=fnal/O=Fermilab/OU=Robots/CN=novagpvm01.fnal.gov/CN=cron/CN=Anna Mazzacane/CN=UID:mazzacan
  - ✓ Open a SNOW request through the your Experiment and Category: 'Batch Submission group' requesting the subject you got in the previous step, to be added to your voms user credentials (example request: RITM0318692).
  - ✓ Add your cron principal in your .k5login

# Backup slides

- Set a script that has the following content:

```
#!/bin/sh
export KRB5CCNAME=/tmp/krb5cc_$(id -u)_kcron
/usr/krb5/bin/kcron
/usr/krb5/bin/kx509
voms-proxy-init -rfc -noregen -cert /tmp/x509up_u$(id -u) -voms
fermilab:/fermilab/nova/Role=Production
scp -qBp /tmp/x509up_u$(id -u) ${USER}@expgpvm0x.fnal.gov:/var/tmp/${USER}.Production.proxy
```

- Make this script run as a cronjob every hour

**The production proxy is ready to use!**